

# Programming Challenge

# C++ Code

```
int main(int argc, char *argv[]){
    //parsing comand line arguments
    if (argc !=3){
        | cerr << "Usage: nowhitespace in out" << endl;
        | return 1;
    }

    //reading infile and writing noWhitespace outfile
    string line;
    ifstream infile( argv[1], std::ifstream::binary );
    ofstream outfile (argv[2], std::ofstream::binary );
    if (outfile) outfile < ""; //cleaning the outfile
    ** →
    ** →
    return 0;
}
```

```
if ( infile && outfile ){
    | while ( infile.good() ){
    | | getline (infile, line);
    | | for (int i = 0; i < line.length(); i++){
    | | | if (line[i] != ' ') outfile << line[i];
    | | }
    | | outfile << endl;
    | }
    | infile.close();
    | outfile.close();
    }
    else cout << "Unable to open file";
```

# Documentation – POD Format

=encoding utf8

=head1 NAME

nowhitespace - Remove whitespaces from a file

=head1 SYNOPSIS

nowhitespace <infile> <outfile>

=head1 DESCRIPTION

Reads from I<infile> and writes same text without whitespaces to I<outfile>.

=head1 EXAMPLE

nowhitespace whiteSpaces.txt noWhitespaces.txt

=head1 AUTHOR

Written by Robert Wagner.

=head1 SEE ALSO

Nothing.

```
NOWHITESPACE(1)      User Contributed Perl Documentation      NOWHITESPACE(1)

NAME
    nowhitespace - Remove whitespaces from a file

SYNOPSIS
    nowhitespace <infile> <outfile>

DESCRIPTION
    Reads from infile and writes same text without whitespaces to
    outfile.

EXAMPLE
    nowhitespace whiteSpaces.txt noWhitespaces.txt

AUTHOR
    Written by Robert Wagner.

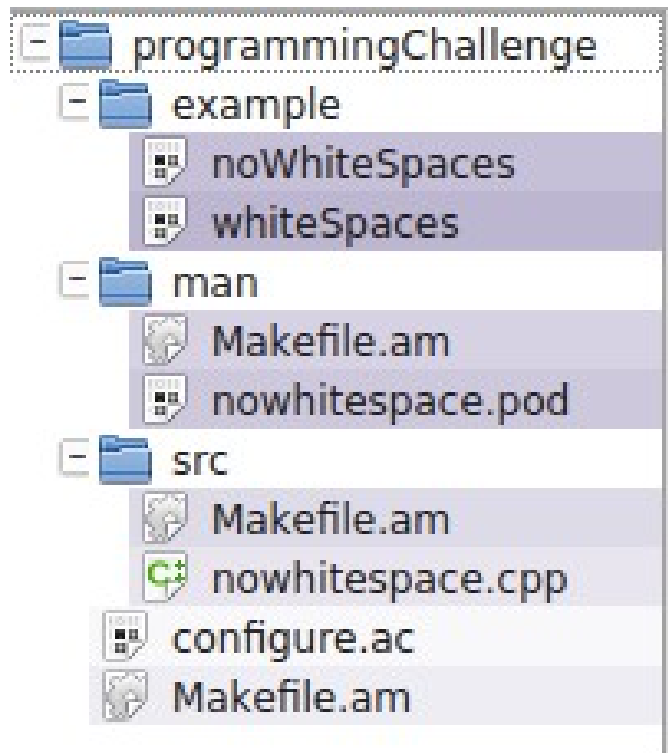
SEE ALSO Nothing.
perl v5.14.2              2013-05-01              NOWHITESPACE(1)
```

# Documentation – Manpage

A Manpage should at least consist of the following sections

- NAME: Name and one-line description
- SYNOPSIS: formal description of how to run the program and the parameters the function takes
- DESCRIPTION: explanation of the tool and the parameters
- EXAMPLES: example calls
- SEE ALSO: List of similar tools

# Automake - Filestructure



- Makefile.am file in each directory that needs instructions
- configure.ac only in basedir

# Makefile.am in basedir

SUBDIRS = src man  
EXTRA\_DIST = example

- SUBDIRS specifies Makefile.am directories
- EXTRA\_DIST just copies all files in the directory into the package

# configure.ac in basedir

```
AC_INIT([nowhitespace], [1.0],[em@i.i])
AM_INIT_AUTOMAKE([-Wall -Werror])
AC_PROG_CXX
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([
    Makefile
    src/Makefile
    man/Makefile
])
AC_OUTPUT
```

- AC\_INIT parameters are: package name, version, and support adress.
- AM\_INIT\_AUTOMAKE parameters: -Wall shows all warnings, -Werror threats all warnings as errors
- AC\_PROG\_ : selection of programm language/compiler (here C++)
- AC\_CONFIG\_HEADERS: file autoconfig uses for configuration
- AC\_CONFIG\_FILES: Makefiles (Makefile.in) that autoconfig is supposed to create
- Final statement

# Makefile.am in src

```
bin_PROGRAMS = nws  
nws_SOURCES=nws.cpp
```

- bin\_PROGRAMS: listing of the programs that are to be compiled  
autotools knows what compiler to use based on configure.ac AC\_PROG\_ parameter
- nws\_SOURCES listing of the source files to use to compile the bin\_PROGRAMS



# Makefile.am in man

```
dist_noinst_DATA = nws.pod
man_MANS = nws.1

nws.1: nwse.pod
    pod2man -c 'User Commands'
-r "$$(VERSION)" $< $@

clean-local:
    rm -rf nws*
```

- `dist_noinst_DATA`: does not copy listed file when 'make install' is called
- `man_MANS`: specifies a Manpage, that is automatically installed

# Programming Challenge - Notes

- Packaging instructions:  
autoreconf --install  
./configure  
make  
make install
- Since package is not declared foreign in configure.ac is expected to adhere the GNU standard. Therefore following files are expected in the base directory:  
README ChangeLog AUTHORS NEWS INSTALL  
COPYING

# Make Challenge

# Makefile.in - Basics

- Syntax:

Target: dependency

<Tab> System calls

<Tab> System calls

- Reference:

Reference= item1 item2 item3

Target: \$(Reference)

\$@ references the current Target

\$< references the current dependency

# Makefile.in - Basics(2)

- Pattern rules:

TargetA: A.dep B.dep

%.dep:

<Tab> System calls (using '\$@' and/or '\$<')

- Substitution:

Reference= item1.a item2.a item3.a

Target: \$(Reference:.a=.b)

# Make Challenge - First Try

```
all: first.name last.name .xxd .chk
```

```
first.name: full.name  
cut -d' ' -f2 full.name > first.name
```

```
last.name: full.name  
cut -d' ' -f1 full.name > last.name
```

```
.xxd: last.xxd full.xxd first.xxd
```

```
last.xxd: last.name  
xxd < last.name > last.xxd
```

```
full.xxd: full.name  
xxd < full.name > full.xxd
```

```
first.xxd: first.name  
xxd < first.name > first.xxd
```

```
.chk: last.chk full.chk first.chk
```

```
last.chk: last.name  
sha1sum < last.name > last.chk
```

```
full.chk: full.name  
sha1sum < full.name > full.chk
```

```
first.chk: first.name  
sha1sum < first.name > first.chk
```

```
clean:  
rm *.chk *.xxd first.name last.name
```

```
help:  
echo "all"  
echo ...
```

# Make Challenge - Pattern Rules

```
all: first.name last.name .xxd .chk
```

```
first.name: full.name  
cut -d' ' -f2 full.name > first.name
```

```
last.name: full.name  
cut -d' ' -f1 full.name > last.name
```

```
.xxd: last.xxd full.xxd first.xxd
```

```
last.xxd: last.name  
xxd < last.name > last.xxd
```

```
full.xxd: full.name  
xxd < full.name > full.xxd
```

```
first.xxd: first.name  
xxd < first.name > first.xxd
```

```
.chk: last.chk full.chk first.chk
```

```
last.chk: last.name  
sha1sum < last.name > last.chk
```

```
full.chk: full.name  
sha1sum < full.name > full.chk
```

```
first.chk: first.name  
sha1sum < first.name > first.chk
```

```
clean:  
rm *.chk *.xxd first.name last.name
```

```
help:  
echo "all"  
echo ...
```

# Make Challenge - Pattern Rules

```
all: first.name last.name .xxd .chk
```

```
first.name: full.name
```

```
cut -d' ' -f2 full.name > first.name
```

```
last.name: full.name
```

```
cut -d' ' -f1 full.name > last.name
```

```
.xxd: last.xxd full.xxd first.xxd
```

```
%.xxd: %.name
```

```
xxd < $< > $@
```

```
.chk: last.chk full.chk first.chk
```

```
%.chk: %.name
```

```
sha1sum < $< > $@
```

```
clean:
```

```
rm *.chk *.xxd first.name last.name
```

```
help:
```

```
echo "all"
```

```
echo ...
```



# Make Challenge - Substitution

```
all: first.name last.name .xxd .chk
```

```
first.name: full.name
```

```
cut -d' ' -f2 full.name > first.name
```

```
last.name: full.name
```

```
cut -d' ' -f1 full.name > last.name
```

```
.xxd: last.xxd full.xxd first.xxd
```

```
%.xxd: %.name
```

```
xxd < $< > $@
```

```
.chk: last.chk full.chk first.chk
```

```
%.chk: %.name
```

```
sha1sum < $< > $@
```

```
clean:
```

```
rm *.chk *.xxd first.name last.name
```

```
help:
```

```
echo "all"
```

```
echo ...
```

# Make Challenge - Substitution

```
SOURCES=first. full. last.
```

```
all: first.name last.name $(SOURCES:.=.xxd) $(SOURCES:.=.chk)
```

```
first.name: full.name
```

```
cut -d' ' -f2 full.name > first.name
```

```
last.name: full.name
```

```
cut -d' ' -f1 full.name > last.name
```

```
%.xxd: %.name
```

```
xxd < $< > $@
```

```
%.chk: %.name
```

```
sha1sum < $< > $@
```

```
clean:
```

```
rm *.chk *.xxd first.name last.name
```

```
help:
```

```
echo "all"
```

```
echo ...
```

# Make Challenge – Final Changes

```
SOURCES=first. full. last.  
  
all: first.name last.name $(SOURCES:!=.xxd) $(SOURCES:!=.chk)  
  
first.name: full.name  
    cut -d' ' -f2 full.name > first.name  
  
last.name: full.name  
    cut -d' ' -f1 full.name > last.name  
  
%.xxd: %.name  
    xxd < $< > $@  
  
%.chk: %.name  
    sha1sum < $< > $@  
  
clean:  
    rm *.chk *.xxd first.name last.name  
  
help:  
    echo "all"  
    echo ...
```

# Make Challenge – Final Changes

```
SOURCES=first. full. last.  
  
all: first.name last.name $(SOURCES:!=.xxd) $(SOURCES:!=.chk)  
  
first.name: full.name  
    cut -d' ' -f2 $< > $@  
  
last.name: full.name  
    cut -d' ' -f1 $< > $@  
  
%.xxd: %.name  
    xxd < $< > $@  
  
%.chk: %.name  
    sha1sum < $< > $@  
  
clean:  
    rm *.chk *.xxd first.name last.name  
  
help:  
    echo "all"  
    echo ...
```